

文章编号: 2095-2163(2021)12-0179-06

中图分类号: TP301

文献标志码: A

基于象群-蚁群算法改进的小车路径规划

柏语蔓, 于莲芝

(上海理工大学 光电信息与计算机工程学院, 上海 200093)

摘要: 社会发展对智能机器人运行要求不断提高,单一蚁群算法已经无法满足路径规划的要求。针对蚁群算法存在探索和开发不平衡的问题,本文提出一种改进蚁群算法。首先采用栅格法建立环境模型;其次使用最佳优先搜索算法预搜索次优路径,来初始化蚁群算法的信息素浓度,以提高算法前期收敛速度;利用象群算法优化蚁群算法参数,以减小参数设置对算法影响,从而提高蚁群算法搜索性能。将改进蚁群算法进行仿真实验,并与传统蚁群算法进行对比。实验结果证明,改进蚁群算法搜索效率更高,迭代次数减少近65%。

关键词: 路径规划; 蚁群算法; 栅格法; 最佳优先搜索算法; 象群算法

Smart car path planning system based on elephant herding and ant colony Optimization

BAI Yuman, YU Lianzhi

(School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)

[Abstract] The requirements for the operation of intelligent robots are constantly increasing with the social development. The ant colony algorithm can no longer meet the requirements of path planning. Aiming at the problem of imbalance in the exploration and development of ant colony algorithm, this paper proposes an improved ant colony algorithm. Firstly, the grid method is used to establish the environment model. Secondly, Best-First Search algorithm is used to pre-search the sub-optimal path to initialize the pheromone concentration of the ant colony algorithm to increase the convergence speed of the algorithm. Then the elephant herding algorithm is used to optimize the parameters of the ant colony algorithm to reduce the effects from parameters setting, thereby improving the search performance of the ant colony algorithm. Finally, the improved ant colony algorithm is simulated and compared with the traditional ant colony algorithm. The experimental results prove that the improved ant colony algorithm has higher search efficiency and reduces the number of iterations by nearly 65%.

[Key words] path planning; ant colony algorithm; grid method; best-first search algorithm; elephant herding algorithm

0 引言

科技发展推动着智能移动机器人领域迅速发展,该领域需要多类技术相互融合^[1],其中路径规划是该领域研究的核心问题^[2]。智能机器人在实际使用中能否高效、安全完成任务,主要取决于路径规划是否合理。目前主流的路径规划算法主要有人工势场法^[3]、模糊逻辑算法^[4]、模拟退火算法^[5]、蚁群算法等。

在众多路径规划算法中,蚁群算法有较好的全局优化能力、鲁棒性、并行性^[6],适合复杂环境中的路径规划问题。该算法由 MarcoDorigo 在 1992 年提出^[7],这是一种模仿蚂蚁觅食的智能路径规划算法。模仿蚂蚁通过释放信息素寻找最优觅食路径,

多轮觅食之后,根据正反馈机制选出信息素浓度最高路线为最优路径。

其中,蚂蚁种群数量、信息素挥发速度、信息素影响因子、启发式影响因子这些参数值选取过大或者过小都会直接影响路径搜索的随机性,进而影响算法收敛速度和全局搜索能力。随机性较弱时,算法的搜索速度加快,但会导致全局搜索能力降低,结果为局部最优。当算法随机性较强时,降低蚁群算法正反馈作用,算法收敛速度变慢,拖慢运行效率。所以如何选择合适的参数是蚁群算法改进的重点。

针对蚁群算法在路径规划应用中存在的问题,目前主流的对蚁群算法的改进方案有:

(1)单一蚁群算法的改进^[8]:单个蚁群算法通过改进算法结构、优化参数、设置信息素初始值和调

作者简介: 柏语蔓(1994-),女,硕士研究生,主要研究方向:智能控制理论与应用;于莲芝(1966-),女,博士,副教授,主要研究方向:数据挖掘、智能控制。

通讯作者: 于莲芝 Email: yulianzhi@163.com

收稿日期: 2021-09-21

整信息素更新原则等方式,提高算法的寻优能力。该改进方案虽然在一定程度上改进了蚁群算法的运行速度、减少陷入局部最优解的可能,但由于大部分参数改动都是需要人为去设定参数,存在算法不灵活的问题。

(2)多蚁群优化算法^[8]:采用不同算法结构、运行机制的蚂蚁群去寻找最优路径,一般主流的运行机制是顺序和并行。该改进方案是通过增加蚁群的多样性,来提高结果为最优解的可能性。其中顺序运行机制需要设置相遇判断准则,而并行运行机制需要制定种群间信息交换机制,并且相遇判断准则和信息交换机制的设计直接影响整个算法的优化结果,甚至会降低蚁群算法的性能。

(3)融合蚁群算法^[8]:通过融合其它优化算法,从初始信息素值、状态转移规则、算法参数、初始路这4个方面优化蚁群算法。该方案可以大大降低蚁群算法的迭代次数,提高寻优速度。通过大量的文献表明,该类改进方案相对于前两种方案具有更强的寻优能力。但是,由于算法的复杂程度增加,会在一定程度上增加算法的运行时间。

本文选用第三类改进方案,将贪婪算法和象群算法融入到蚁群算法之中。针对蚁群算法初始信息素值相同导致的初次迭代过于随机,影响算法的迭代速度的问题,本文在蚁群算法之前加入贪婪算法,选出次优路径,以此为依据设置初始信息素值。针对蚁群算法凭借经验设计参数,导致参数设计不当影响算法性能的现象,本文将象群算法融入到蚁群算法之中,并在象群算法的基础上加入评价函数,为蚁群算法选取合适的算法参数,以此降低算法迭代次数,提升蚁群算法寻优能力。最后将改进后的蚁群算法带入到已知运行环境的机器人小车路径规划实验中,在创建的运行环境栅格图的基础上,用MATLAB进行改进后和初始蚁群算法的路径规划对比仿真实验,根据实验数据分析改进后的算法优劣。

1 栅格法创建环境模型

本文研究的是基于蚁群算法的全局路径规划,即已知环境中障碍物信息。在该情况下,对全局环境进行建模是路径规划的基础和前提。环境模型需要包含环境中所有的已知信息,并且该环境模型描述的方式要方便计算机去存储和运算,满足环境信息随机器人位置变换而随时更新的要求。同时,在环境模型上搜索的最优路径,能让移动机器人在实际运行中少与障碍物发生碰撞。由于栅格法能够满

足环境建模的要求,并且易于创建便于维护,因此本文选择栅格法创建环境模型。

栅格法用坐标系表示机器人运动环境。首先在运行环境边界找一点为坐标原点,并创建坐标系,划分运行环境为等大小的栅格,障碍物等比例映射到栅格图中,进行障碍物标注,由黑白两色分别标记栅格图中的障碍物和自由栅格。

在路径规划仿真实验中,一般不考虑移动机器人的大小和形状,只将其看作一个质点,为了避免实际实验中移动机器人与障碍物发生碰撞,在进行栅格法创建环境模型时,一般会膨胀处理栅格中的障碍物,即将障碍物的边长扩大半个栅格的长度,以此提高移动机器人在实际运动中的安全性。栅格法创建的环境模型如图1所示。

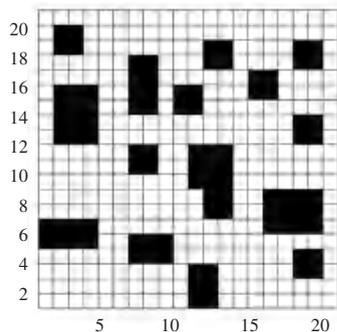


图1 栅格地图

Fig. 1 Raster map

在栅格图中可见,每一块栅格都会由序号*i*和坐标(*x*,*y*)共同去表示。序号是用从1开始的自然数按照从左到右、从上到下的方式去逐一标注每一块栅格,坐标是用创建的XOY坐标系,来表示每一个栅格的坐标。序号和坐标之间的转换关系为:

$$\begin{cases} x = \text{mod}(i, N) - 0.5 \\ y = N + 0.5 - \text{ceil}(\frac{i}{N}) \end{cases} \quad (1)$$

式中:*mod*表示 $\frac{i}{N}$ 的余数;*ceil*表示 $\frac{i}{N}$ 的整数;*N*表示每列栅格数。

2 蚁群算法改进

2.1 初始信息素改进

由于初始信息素浓度相同会导致蚂蚁初次搜索路径过于随机,从而使算法收敛速度减慢。因此本文通过差异化设置栅格图各块信息素浓度的方式,使初始信息素浓度按一定规则分布,降低蚁群初次迭代的随机性,以此缩短收敛时间,提高蚁群算法的

效率。

本文选用最佳优先搜索算法 (Best-First Search, BFS), 在蚁群算法运行之前选择一条次优路径。BFS 算法是一种启发式搜索算法^[9], 有别于穷举类型的搜索算法, 不需要逐个搜索所有的点, 运行效率更高, 更适合复杂、面积较大的运行环境, 适合用作寻找环境中的次优路径。BFS 找出的次优路径如图 2 所示。

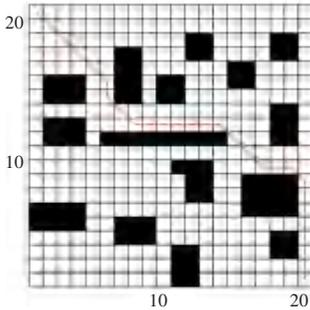


图 2 次优路径图

Fig. 2 Suboptimal path diagram

以 BFS 算法选择的路径结果为依据, 初始化路径中的信息素浓度。初始信息素浓度以次优路径为中心向两边呈标准正态分布, 如公式(2):

$$y = a \frac{1}{\sqrt{2}} \exp\left\{-\frac{x^2}{2\sigma^2}\right\} \quad (2)$$

式中: x 表示点到次优路径的垂直距离, $a \frac{1}{\sqrt{2}}$ 的值为次优路径信息素浓度值。

2.2 参数改进

在蚁群算法中, 参数对算法结果的准确度起着至关重要的作用。原始蚁群算法是凭借实验者的经验来选取参数值^[10], 使用象群算法较强的寻优能力选取蚁群算法中的重要参数。象群算法 (Elephant Herding Optimization, EHO)^[11] 是一种新元启发式优化算法, 较其它优化算法公式简单、便于理解, 具有收敛速度快寻优能力强等优点。

象群优化算法是根据象群的游牧行为启发得来的^[12]。大象成群生活由多个氏族在一位最优秀母象女部长的领导下组成, 每个氏族由母象和她的小象或某些有血缘关系的母象和小象组成, 氏族里也会选出优秀的母象担任族长, 成年后公象会离开群体独自生活, 虽然公象远离其家族, 但其可以通过低频振动与家族中的大象保持联系。

为了使用大象的放牧行为解决各种全局优化问题, 对其做以下理想化规则。

(1) 一个象群部落由多个氏族和一个最优母象女族长组成, 每个氏族都有固定数量的大象。

(2) 每代中每个氏族都会选出一个优秀女族长, 并有固定数量公象离开, 且产生其相同数量的新小象。

象群算法分为两个阶段, 分别是氏族更新阶段和分离阶段。具体过程如下:

2.2.1 氏族更新阶段

氏族里每头大象的位置都会受到女族长的影响, 对于氏族 ci 中的大象 j , 位置更新如公式(3):

$$x_{ci,j}^{new} = x_{ci,j} + \phi(x_{best,ci} - x_{ci,j})r \quad (3)$$

式中: $\phi \in [0, 1]$ 为比例因子; $x_{ci,j}^{new}$ 为氏族 ci 中大象 j 的新位置; $x_{ci,j}$ 为氏族 ci 中大象 j 的旧位置; $x_{best,ci}$ 为氏族 ci 女族长; $r \in [0, 1]$ 为随机数。

女族长位置更新位置如公式(4):

$$x_{center,ci}^{new} = \delta \times x_{center,ci} \quad (4)$$

式中: $\delta \in [0, 1]$ 为 $x_{center,ci}$ 对 $x_{best,ci}$ 的影响因子, $x_{best,ci}$ 为氏族 ci 的中心位置。其中, $x_{center,ci,d}$ 的表达式如公式(5):

$$x_{center,ci,d} = \frac{1}{n_{ci}} \times \sum_{j=1}^{n_{ci}} x_{ci,j,d} \quad (5)$$

式中: $d \in [1, D]$ 为维度; D 为最大维度; n_{ci} 为氏族 ci 的大象总数。

2.2.2 分离阶段

为了模拟成年后的公象离开群体独自生活这一过程, 并获得更大搜索能力, 假设适应度最差的大象个体, 在每代中执行分离算子, 如公式(6):

$$x_{worst,ci} = x_{min} + (x_{max} - x_{min} + 1) \times r \quad (6)$$

式中: x_{max} 和 x_{min} 分别为大象个体位置上界和下界, $x_{worst,ci}$ 为 ci 氏族中最差大象个体。

改进方案是将信息素挥发速度、信息素影响因子、启发式影响因子和蚂蚁种群数量这 4 个参数, 作为象群的位置信息, 创建一个四维的坐标矩阵。由象群算法不断迭代更新象群位置, 选出最优的母象位置, 同时设计评价函数来鉴定所选参数的合理性, 以此使象群算法得出的参数值可以更好的适应蚁群算法。

设计评价函数的目的, 是评估象群算法选取参数是否适合蚁群算法。所以评价函数设计过程需要考虑到蚁群算法的几个重要性能指标。这些指标分别为寻优能力、稳定性、收敛速度、运算时间, 该评价

函数设计如公式(7)~(9):

$$Fit_k(s) = \omega_1 L_k(s) + \omega_2 S_k(s) + \omega_3 E_k(s) + \omega_4 T_k(s) \quad (7)$$

$$L_k(s) = f\left(\frac{1}{L_{local}}; 0; 1\right) \quad (8)$$

$$S_k(s) = \frac{1}{L_{std} + 1} \quad (9)$$

式中: $Fit_k(s)$ 为将当前大象 k 的位置矩阵作为参数,运行蚁群算法后的适应值; ω_1 、 ω_2 、 ω_3 、 ω_4 为寻优能力、稳定性、收敛速度和运算时间各指标对应的权值,权值相加为 1; $L_k(s)$ 为使用当前大象位置作为参数的蚁群算法的寻优能力; f 为正态分布函数; L_{local} 表示此时蚁群算法的最优解; $S_k(s)$ 为此时蚁群算法稳定性; L_{std} 为每次迭代所得最短距离方差; $E_k(s)$ 表示此时蚁群算法寻找到最优解所需迭代次数; $T_k(s)$ 表示此次蚁群算法运算时间。

2.3 算法实现

该算法的主要实现步骤如下:

Step 1 使用栅格法创建环境模型;

Step 2 贪心算法选出次优路径,并根据路径选择结果,初始化各节点之间的信息素浓度;

Step 3 初始化象群算法,给各个大象随机选取初始位置和速度,并为比例因子 ϕ 和影响因子 δ 赋值;

Step 4 将每个大象的位置信息作为参数带入蚁群算法中,计算在各参数下运行蚁群算法获得的 L_{local} 、 L_{std} 、 E_k 、 T_k 。

Step 5 将步骤 4 中每个大象对应的 $L_k(s)$ 、 $S_k(s)$ 、 $E_k(s)$ 和 $T_k(s)$ 值带入适应度函数中,算出各个大象的适应度值;

Step 6 比较各大象的适应度值,选取适应度最优的大象为族长,同时选出适应度最差的大象;

Step 7 用公式(4)更新族长的位置,用公式(5)更新适应度较差大象的位置,用公式(3)更新其余大象的位置;

Step 8 判断象群算法的迭代次数是否达到设定的最大值,若为最大值,就将未更新位置前族长的位置输出;否则,回到步骤 4;

Step 9 用输出的最优位置为参数,带入到蚁群算法中,用蚁群算法选出最优路径,得到全局最优解。

改进后蚁群算法流程如图 3 所示:

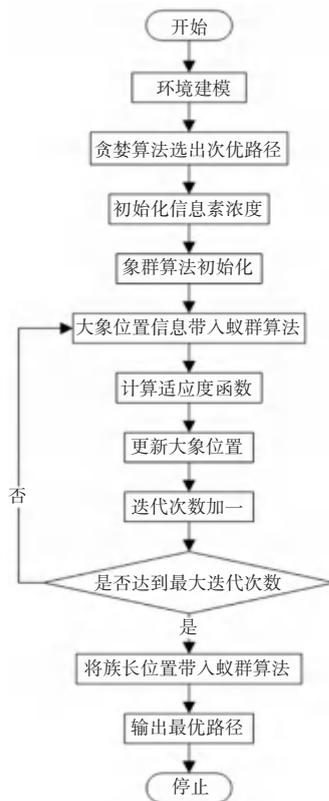


图 3 改进后蚁群算法流程图

Fig. 3 Flow chart of improved ant colony algorithm

3 仿真结果与分析

本文选用的仿真环境为 MATLAB2018b,环境模型为 $20 * 20$ 的栅格。对比改进前后的蚁群算法的仿真结果,验证改进算法的可行性和优越性。

在信息素加强系数都为 1 的情况下,根据文献^[13]给出的传统蚁群算法参数,其中蚂蚁的数量为 40、信息素挥发速度为 0.3、信息素影响因子为 5、启发式影响因子为 2。通过象群算法改进后的蚁群算法参数选择见表 1,次优路径的信息素浓度值为 8。将两种情况,进行 100 次迭代, 20 次重复实验,比较两者的性能。表 2 给出了运行 20 次象群算法后选取最佳优化结果所对应的优化蚁群算法参数。

表 1 算法参数表

Tab. 1 Algorithm parameter

参数类型	数值
种族中大象个数	10
比例因子 ϕ	0.5
影响因子 δ	0.1
加权系数 ω_1	0.7
加权系数 ω_2	0.1
加权系数 ω_3	0.1
加权系数 ω_4	0.1
最大迭代次数	100
重复试验次数	20

表 2 优化参数值

Tab. 2 Optimized parameter values

参数	蚂蚁的数量	信息素挥发速度	信息素影响因子	启发式影响因子
参数值	48	0.412 6	3.593 4	6.238 6

由图 4、5 结果显示,采用传统蚁群算法和改进后的蚁群算法,在路径规划上结果具有一定差异。通过对比图 6、7 的结果(即改进前后的收敛曲线)可以看出,改进后的蚁群算法结果更优,收敛更快,且算法运行稳定性更高。因此,可以预测该改进方案在复杂的运行环境中优势会更加明显。

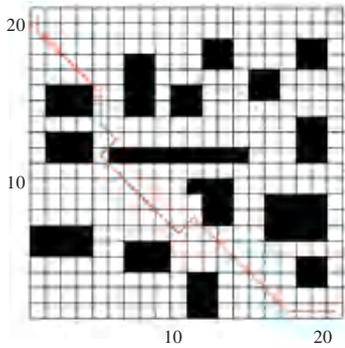


图 4 传统蚁群算法仿真结果

Fig. 4 Simulation results of traditional ant colony algorithm

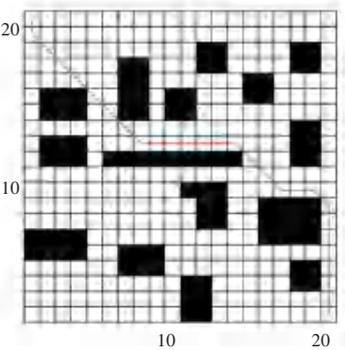


图 5 改进后蚁群算法仿真结果

Fig. 5 Simulation results of the improved ant colony algorithm

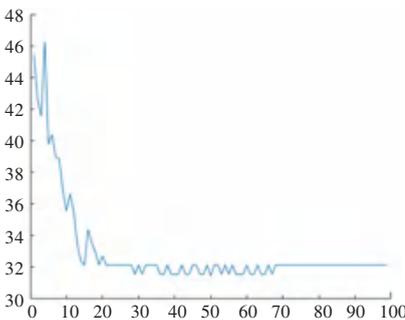


图 6 传统蚁群算法收敛曲线图

Fig. 6 Convergence curve of traditional ant colony algorithm

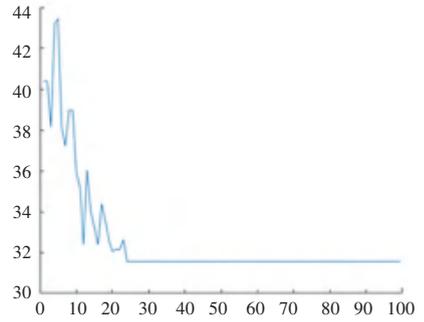


图 7 改进后蚁群算法收敛曲线图

Fig. 7 Convergence curve of improved ant colony algorithm

传统蚁群算法和改进后蚁群算法的仿真数据对比结果见表 3。从这些数据中可以清晰看出,在路径长度上,改进算法路径更优,且其到达收敛时的迭代次数远远少于改进之前。改进的蚁群算法运行时间长于传统蚁群算法,是因为该时间包含了象群算法的运行时间,由于在移动小车运行前进行路径规划,所以满足本文实验要求。从 20 次实验路径长度的平均值和方差可以看出,改进后的算法更加稳定。由此证明,使用象群算法选择参数,在算法收敛速度、稳定性和结果精确度上会优于凭借经验选取参数。

表 3 仿真结果对比

Tab. 3 Comparison table of simulation results

算法	最优路径长度/cm	平均值/cm	方差	迭代次数	运行时间/s
传统蚁群算法	32.321 3	33.526 2	17.029 7	69	5.141 9
改进蚁群算法	31.713 7	32.128 3	8.439 9	24	20.014

4 结束语

本文研究了基于蚁群算法改进的移动机器人路径规划,改进措施针对蚁群算法在寻优时收敛速度不够快^[14],并且因参数选择不当而导致结果陷入局部最优解的情况^[15]。在使用栅格法创建环境地图的基础上,使用贪婪算法找到次优路径,以此为依据初始化信息素浓度,提高蚁群算法的收敛速度;加入象群算法解决蚁群算法凭借经验选择参数的问题,同时用蚁群算法的各项指标设置了象群算法的适应度函数,让象群朝着适应度值高的位置移动,以此选出适合的蚁群算法参数,解决蚁群算法因为参数选择不当而导致的局部最优解问题。仿真结果证明,该改进方案是具有可行性的,并且有助于蚁群算法快速选取全局最优路径。

(下转第 189 页)